

**BACKGROUND OF THE INVENTION**Field of the Invention

5 The present invention relates to a digital signal processor capable of efficiently carrying out the arithmetic or interruption processing mainly of successive signals at a high speed through a small number of steps.

Description of the Prior Art

10

Fig. 1 is a block diagram showing the constitution of DSSP1 (Digital Speech Signal Processor 1), namely, an exemplary conventional digital signal processor, "A High-speed VLSI Signal Processor with Normalizing Floating-point Systems", the proceedings of the annual communication Symposium of the Institute of Electronics and Communication Engineers of Japan (IECEJ), 1986, Japan.

15 Shown in Fig. 1 are program counter (PC) 1 internally provided with a stack for instruction address control, an instruction mask ROM 2 storing microinstructions, an instruction register (IR0) 3 for receiving one microinstruction provided by the instruction mask ROM 2 or one external microinstruction every machine cycle, an instruction register (IR1) 4 for receiving only the bit field requiring decoding included in a microinstruction given to the instruction register (IR0) 3, an instruction decoder 5 for decoding the microinstruction given to the instruction register (IR1) 4, a program bus (P-Bus) 6 for distributing microinstructions to the functional units, a register (BI) 7 which receives immediate data (18-bit width) included in a microinstruction provided on the program bus (P-Bus) 6 and applies the same to a data bus (D-Bus) 8 (18-bit width) for internally transferring data obtained by operation, a register (AM) 9 which receives an address mode instruction of data memory through the program bus (P-Bus) 6, a register (AD) 25 10 (4w x 16-bit width) for holding address pointer information for generating an indirect address, a page register (PR) 11 (3-bit width) which specifies the page in an external data memory, an address computation unit (AAU) 12 (9-bit width) capable of simultaneously generating three addresses at the maximum, an address register (AR0) 13, an address register (AR1) 14, an address register (AR2) 15, an address selector (RAS) 16, a loop counter (LC) 17, a status register (SR) 18 for indicating the operating mode and status of the processor, a DMA control unit 19 for controlling the direct data transfer between serial I/O ports (SI0/1, SO0/1) 32 and an external data memory, an address register (AR) 20 for holding addresses of 12-bit width to be given to an external data memory, a dual port internal data memory (20-RAM) 21 of 512w x 18 bits capacity capable of simultaneous read and write of two data, a register (DP0) 22 for holding input data of operand, a register (DP1) 23 for holding input data of operator, a multiplier (FMPL) 24 for multiplying the floating point of 12E6 bit format, a register (P) 25 for holding the results of operation of the multiplier (FMPL) 24, a selector 26, a selector 27, a floating-point arithmetic logical operation unit 28 mainly for carrying out the floating-point operation of 12E6 bit format, accumulators (ACC0 to ACC 3) 29 of 4w x 18 bits for holding and accumulating the outputs of the floating-point arithmetic logical operation unit (FALU) 28, a data register (DR) 30 connected to the data bus (D-Bus) 8 to temporarily hold data to be read from and to be written in an external data memory, a read/write control circuit (R/W Cont) 31 for reading data from and writing data in an external data memory, serial I/O ports (SI0/1, SO0/1) 32 for full-duplex two-channel data transfer with external devices, an interrupt control circuit (Int Cont) 33, an external data memory bus control circuit (Bus Cont) 34, a clock control circuit (CLK Cont) 35 for controlling internal timing, and a selector 36.

45 Fig. 2 is a time chart of assistance in explaining the microinstruction execution sequence of the digital signal processor DSSP1 shown in Fig. 1. Shown in Fig. 2 are cycle timing 40 consisting of four phases of clocks, fetch stage timing 41 showing stages of the address output of the program counter (PC) 1 and the microinstruction input of the instruction register (IR0) 3, decode stage timing 42 of decoding the input microinstruction of the instruction register (IR1) 4 by the instruction decoder 5, timing 43 of updating the address computation unit 12 in the decode stage, timing 44 of operation of the floating-point multiplier (FMPL) 24, timing 45 of operation of the floating-point arithmetic logical operation unit (FALU) 28, timing 46 of transferring data through the data bus (D-Bus) 8 between the registers, and timing 47 of reading data from and writing data in the external data memory through the data register (DR) 30.

Referring to Fig. 3 showing the respective constructions of microinstructions of 32-bit width per word 55 representing four groups of microinstructions of the digital signal processor DSSP1 of Fig. 1, indicated at 50 is a sequence instruction for controlling instruction processing steps, at 51 is a mode instruction for initializing and setting modes of the status register (SR) 17, the address computation unit (AAU) 12 and the DMA control unit 19, at 52 is an operation instruction mainly for controlling the operation of the floating-point

arithmetic logical operation unit (FALU) 28 and parallel data transfer accompanying the operation of the floating-point arithmetic logical operation unit (FALU) 28, and at 53 is a load instruction for loading immediate data on an optional register or a data memory.

The operation of the digital data processor (DSSP1) will be described hereinafter, in which the components will be denoted by the abbreviated designation shown in the foregoing description.

First the general mode of operation will be described with reference to Fig. 1. In this digital signal processor DSSP1, the P-Bus 6 and the D-Bus 8 are provided individually. The application of the microinstruction to the IRO 3, the transfer of the microinstruction through the P-Bus 6, the decoding of the microinstruction by the instruction decoder 5, and execution of the instruction by the D-Bus 8, the FMPL 24 and the FALU 28 are carried out in parallel through a pipeline process. The D-Bus 8 and all the execution units including the 2P-RAM 21 are of register-base, namely, all the inputs and outputs are connected to the registers. In the timing of access to the registers, outputs are provided at the leading edge of the machine cycle and the outputs are set in the registers at the trailing edge of the machine cycle. The contents of data actually processed is not the contents of data set in the register by the same microinstruction, but the contents of data set in the register by the preceding microinstruction. Such a mode of operation is called as delayed operation. The interior of the arithmetic unit is partitioned into sections by the registers to enable the parallel operation of the sections. For example, the FMPL 24 continually executes the floating-point multiplication once every machine cycle. In applying data to the FMPL 24, the data is set in the DP0 22 and DP1 23 by the preceding microinstruction, and the contents of the P 25 is fetched by the succeeding or later microinstruction to obtain the results of multiplication. While the contents of the P 25 is being fetched, the data is held by the DP0 22, DP1 23 and the P 25. Accordingly, one multiplication operation formerly requiring three microinstructions for data input, multiplication and data output can be carried out by one microinstruction when the process is executed continuously.

In this DSSP1, the FMPL 24 and the FALU 28 are connected by the P 25. The FALU is able to accumulate the contents of the P25 in the ACC0 29 to the ACC3 29 to execute one term of product-sum operation which is often used in filtering and the butterfly operation of fast Fourier transform (FFT) in one machine cycle similarly to one pair of a multiplier and accumulator shown in "Packing a signal processor onto a single digital board", Louis Schirm, Electronics, Dec. 20, 1979. For example, the product-sum is calculated by using an expression:

$$\sum_{i=1}^N = (a_i \times b_i)$$

where N is an integer not less than 1 (one), and  $a_i$  and  $b_i$  are input data. In this processor, three microinstructions for applying data to the DP0 22 and DP1 23, for executing multiplication by the FMPL 24 and for accumulating the results of multiplication set in the P 26 by the FALU 28 in the ACC0 29 to obtain one term of the product-sum. Naturally, when the operation is carried out continuously, one term of the product-sum can be obtained for one microinstruction. Thus, to obtain one term of the product-sum for one microinstruction, two input data corresponding to the input data  $a_i$  and  $b_i$  must be given respectively to the DP0 22 and the DP1 23 every one microinstruction. Accordingly, the 2P-RAM 21 is enabled to supply the two input data, and a bus is provided to transfer the data read from the 2P-RAM 21 directly to the DP0 22 and DP1 23 without using the D-Bus 8 to avoid bus contention in the D-Bus 8. The AAU 12 has output means for selectively providing two address data among address data of 9-bit width provided through the AR0 12, AR1 14 and AR2 15 mainly to address the input data of the 2P-RAM 21. The AAU 12 is able to specify three addresses simultaneously at the maximum only in generating addresses for the two input data given thereto from the 2P-RAM 21, and an address for one output data given through the DR 30 and the AR 20 to the external data memory. Each addressing is so-called indirect addressing system using an address pointer internally set in the AAU 12. The AR0 13 is susceptible to increment, modulo, bit reverse, repeat, increment base address and updating of increment, while the AR1 14 and the AR 15 are susceptible only to simple increment. The AAU 12 is able to perform address operation only in a 9-bit natural binary system. In specifying a 12-bit address in the external data memory, three bits for specifying a memory page are added to the nine bits to specify twelve bits.

On the other hand, since the FMPL 24 and the FALU 28 execute operation in a normalized floating-point system of 12E6, all the data for the 2P-RAM 21, the DP0 22, the DP1 23, the ACC0 29 through ACC3 29, the DR 30, the D-Bus 8 and BI 7 are of 18-bit width, and hence the FALU 28 needs a special operation



Europäisches Patentamt  
European Patent Office  
Office européen des brevets



Publication number: **0 554 917 A2**

**EUROPEAN PATENT APPLICATION**

(21) Application number: 93104194.1

(51) Int. Cl.<sup>5</sup>: G06F 9/26, G06F 15/78

(22) Date of filing: 01.06.88

This application was filed on 16 - 03 - 1993 as a divisional application to the application mentioned under INID code 60.

(30) Priority: 05.06.87 JP 140872/87  
27.07.87 JP 186858/87  
06.08.87 JP 197009/87  
29.10.87 JP 273763/87  
30.10.87 JP 274810/87  
25.11.87 JP 296611/87  
25.11.87 JP 296612/87  
15.12.87 JP 316553/87

(43) Date of publication of application:  
11.08.93 Bulletin 93/32

(46) Publication number of the earlier application in accordance with Art.76 EPC: 0 293 851

(86) Designated Contracting States:  
DE FR GB IT NL SE

(71) Applicant: MITSUBISHI DENKI KABUSHIKI  
KAISHA  
2-3, Marunouchi 2-chome Chiyoda-ku  
Tokyo 100(JP)

(72) Inventor: Murakami, Tokumichi, c/o  
MITSUBISHI DENKI K.K.  
Communication Sys. Dev. Lab., 1-1, Ofuna  
5-chome

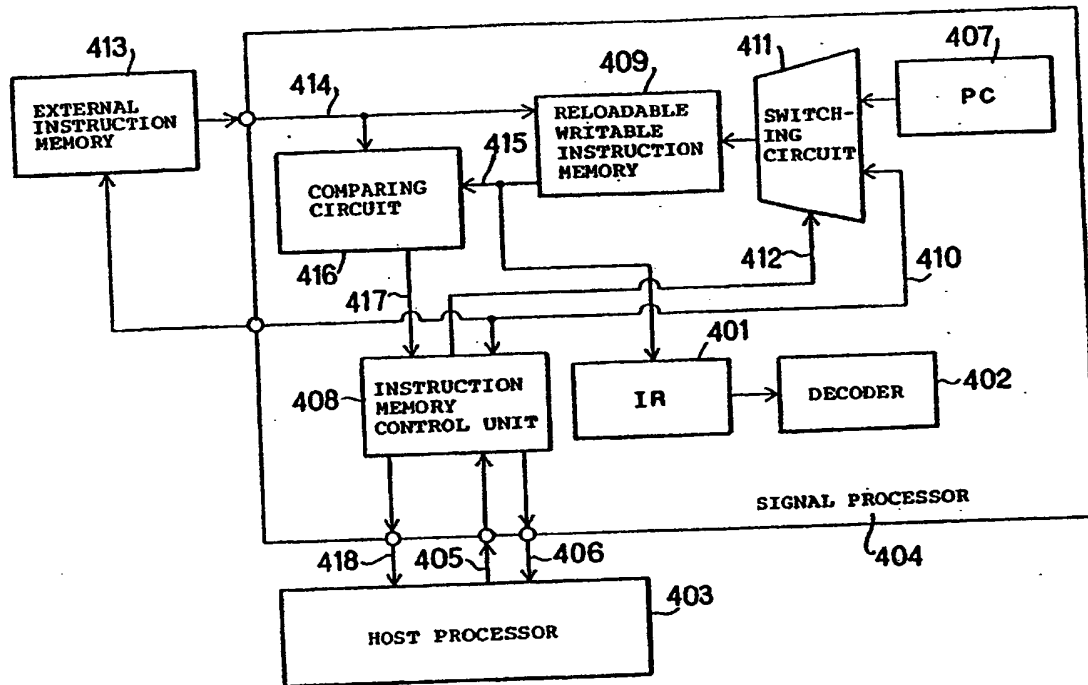
Kamakura-shi, Kanagawa(JP)  
Inventor: Kamizawa, Koh, c/o MITSUBISHI  
DENKI K.K.  
Communication Sys. Dev. Lab., 1-1, Ofuna  
5-chome  
Kamakura-shi, Kanagawa(JP)  
Inventor: Katoh, Yoshiaki, c/o MITSUBISHI  
DENKI K.K.  
Communication Sys. Dev. Lab., 1-1, Ofuna  
5-chome  
Kamakura-shi, Kanagawa(JP)  
Inventor: Ohira, Hideo, c/o MITSUBISHI DENKI  
K.K.  
Communication Sys. Dev. Lab., 1-1, Ofuna  
5-chome  
Kamakura-shi, Kanagawa(JP)  
Inventor: Kameyama, Masatoshi, c/o  
MITSUBISHI DENKI K.K.  
Info. Sys. and Elec. Dev. Lab., 1-1, Ofuna  
5-chome  
Kamakura-shi, Kanagawa(JP)  
Inventor: Kinjo, Naoto, c/o MITSUBISHI DENKI  
K.K.  
Communication Sys. Dev. Lab., 1-1, Ofuna  
5-chome  
Kamakura-shi, Kanagawa(JP)

(74) Representative: Eisenführ, Speiser & Partner  
Martinistrasse 24  
W-2800 Bremen 1 (DE)

(54) Digital signal processor.

(57) A digital signal processor of a simple circuit configuration capable of implementing arithmetic processes and interruption processes efficiently in a reduced number of steps at a high processing speed. The digital signal processor comprises instruction execution pipeline stages including a stage in which data is read from a data memory and the data is applied to an arithmetic unit; an arithmetic unit for the execution stage, including a barrel shifter, a multiplier and an arithmetic and logic unit, a normalizing barrel shifter, a round-off/accumulation adder, internal data memories and a DMA transfer bus for a write/accumulation stage, an address generating unit capable of parallel and two-dimensional generation of two inputs one output data memory addresses and a DMA control unit for controlling the two-dimensional data transfer through a DMA bus between the internal data memories and an external data memory for an instruction execution stage.

Fig. 38



## EP 0 554 917 A2

mode for calculating a special address initial value. Accordingly, the data representing the result of operation stored in the AR0 13, the AR1 14, the AR2 15, the AR 20 and the ACC0 29 through the ACC3 29 are not compatible with those data.

The DMA control unit 19 controls, independently of the microinstruction, full-duplex 2-channel data transfer between the serial I/O ports SI0/1 32 and SO0/1 32, and the external data memories through the D-Bus 8 and the AR 20 and DR 30. Therefore it is possible that the microinstruction operation controlled by the instruction decoder 5 and the internal resource contend with each other.

To avoid the contention, the instruction decoder 5 is held inoperative for six machine cycles for every word to interrupt operation according to the microinstruction in transferring data by the DMA control unit 19.

After all, the DSSP1 is capable of performing the following operations in parallel within one microinstruction in executing microinstructions:

- (1) The 9-bit address operation of three kinds at the maximum by the AAU 12.
- (2) The floating-point multiplication of 12E6 by the FMPL24.
- (3) The floating-point operation of 12E6 by the FALU 28.
- (4) Data transfer through the D-Bus 8 and the DR 30 between the external memories.
- (5) DMA data transfer through the full-duplex 2-channel serial I/O ports SI0/1 32 and SO0/1 32, D-Bus 8 and the DR 30 between the external memories.

The microinstruction execution timing of the DSSP1 will be described with reference to Fig. 2. The machine cycle 40 of the DSSP1 is divided into four phases of timing P0 through P3. The nominal machine cycle time is as high as 50 nsec. Accordingly, it is practically difficult to accomplish three operations, namely, reading a microinstruction from the instruction mask ROM 2, decoding the microinstruction by the instruction decoder 5 and execution of the instruction by the internal resources such as the FMPL 24 and the FALU 28, within one machine cycle. Accordingly, the three operations are divided into stages for each machine cycle to form a three-stage pipeline to enable high-speed operation. The following operations are performed in the stages of the three-stage pipeline.

### (1) Fetch stage 41:

A microinstruction address is provided by the PC 1, a microinstruction is read from the instruction mask ROM 2 and the microinstruction is set in the IRO 3.

### (2) Decode stage 42 and 43:

The microinstruction is transferred from the IRO 3 to the IR1 4, the microinstruction is decoded by the instruction decoder 5, the program control mode is set, the microinstruction is transferred from the IRO 3 to the P-Bus 6, and address operation of the AAU 12 through the AM 9 and the AD 10.

### (3) Execution stage 44, 45, 46 and 47:

Operation of data by the FMPL 24 and FALU 28, data transfer through the D-Bus 8, and access to the external data memories through the AR 20 and the DR 30.

Thus, the DSSP1 needs three machine cycles to execute one microinstruction. This processor executes one microinstruction equivalently by the pipeline method. Accordingly, the actual execution of the microinstruction is delayed by two machine cycles from the read of the microinstruction from the instruction mask ROM 2. To avoid the timing contention between the internal resources, the internal buses are divided into the P-Bus 6 and the D-Bus 8, and the instruction mask ROM 2 and the 2P-RAM 21 are separated. However, since a branch instruction is executed actually in the decode stage [article (2)], the microinstruction being set in the IRO 3 is executed in the decode stage. That is, an instruction succeeding a branch instruction is executed unconditionally. To avoid such unconditional execution of instructions, the DSSP1 changes the instruction succeeding the branch instruction automatically into a no-operation instruction (NOP) during the execution of the branch instruction. Such a function is aimed at simplifying microinstruction description, however, one machine cycle is wasted in the branching operation and two machine cycles are wasted in indirect branching operation using the D-Bus 8. Generally, no problem arises in about 80% of unconditional branching operations even if the succeeding instruction is executed when the sequence of instruction description is arranged properly and hence loss of machine cycles can be avoided. However, the DSSP1 is unable to avoid the loss of machine cycles.

The microinstruction set of the DSSP1 will be described hereinafter with reference to Fig. 3. A microinstruction set includes only four instructions, namely, a sequence instruction, a mode instruction, an

operation instruction and a load instruction.

The sequence instruction controls the PC 1 for loop and subroutine call. The mode instruction initializes and sets the modes of the AAU 12, the selector 16, the LC 17, the SR 18 and the DMA control unit 19. The load instruction is used for loading immediate data of 18-bit width on the registers connected to the D-Bus 8 through the BI 7. The objective resources of the foregoing three microinstructions are fixed depending on instruction operation. On the other hand, as regards the operation instruction, all the internal resources capable of parallel operation must be specified directly. Accordingly, the bit length of the instructions is dependent on the bit length of the operation instruction. The DSSP1 uses horizontal microinstructions of 32-bit width. The FMPL 24 is free to run and no instruction is given thereto directly. The operation of the FALU 28 is specified directly by an instruction. For example, operations of the FALU 28 is controlled by the following instructions.

- (1) Absolute value instruction:  $|X|$
- (2) Signum function instruction:  $\text{Sign}(Y) \cdot X$
- (3) Addition instruction:  $X + Y$
- (4) Subtraction instruction:  $X - Y$
- (5) Maximum value instruction:  $\text{MAX}(X, Y)$
- (6) Minimum value instruction:  $\text{MIN}(X, Y)$
- (7) Fixed-to-floating translation instruction:  $\text{FLT}(X)$
- (8) Floating-to-fixed translation instruction:  $\text{FIX}(X)$
- (9) Shift instruction:  $R1, L1 \text{ to } L8$
- (10) Logic instruction: AND, OR, EOR, NOT
- (11) Mantissa addition instruction:  $X_M + Y_M$
- (12) Characteristic subtraction instruction:  $X_E - Y_E$

However, it is a problem in the DSSP1 that the operation of the DSSP1 is based on floating-point operation, while the DSSP1 carries out logic and address operations on the basis of fixed-point operation. As mentioned above, the floating-point operation and the fixed-point operation are not compatible with each other. In addressing in the memory, for example, on the basis of the results of operation, the instruction (8) must be executed by the FALU 28. Furthermore, since floating-point data is not handled in general data input and output operation, the instruction (7) or the instruction (8) must be executed for every data input or output operation to translate the data.

Another problem in the DSSP1 is that bits are always truncated in normalizing floating-point data entailing errors in operation, because the accuracy of operation of the signal processor is limited. However, when floating-point data is normalized only by truncating bits, the absolute value of the results of operation is always smaller than the true value and hence the distribution of the errors is not random. The errors can be regarded as negligibly small by increasing the operation word length. However, since the ordinary signal processor is required to operate at a high speed, increase in the operation word length is limited.

Such a problem cannot be ignored particularly in image signal processing in which interframe processing is performed by an IIR digital filter (recursive digital filter), and the DSSP1 must round the results of operation by a logical operation instruction or the like. Furthermore, in a general signal processing algorithm, in most cases, the accuracy of operation is regulated specifically for every unit process, and hence the accuracy of operation does not necessarily coincide with the operation word length of the signal processor. In such a case, the format of the operation data is converted repeatedly by the FALU 28 for every unit process.

It is a further problem in the DSSP1 that operation capable of high-speed processing is limited only to product-sum operation. Such a limitation to the mode of operation is not a problem for FFT and FIR filters. However, the recent signal processing algorithm requires for operation to determine the degree of approximation of vectors  $\vec{A}$  and  $\vec{B}$ , i.e., distance calculation, such as expressed by the following expressions to be carried out at a high processing speed.

**BEST AVAILABLE COPY**

**CLIPPEDIMAGE= JP401037124A**

**PAT-NO: JP401037124A**

**DOCUMENT-IDENTIFIER: JP 01037124 A**

**TITLE: PULSE WIDTH MODULATING SIGNAL GENERATOR**

**PUBN-DATE: February 7, 1989**

**INVENTOR-INFORMATION:**

**NAME**

**KUBO, KENJI**

**WATANABE, MASAHIKO**

**OMAE, TSUTOMU**

**HANEI, HIROYUKI**

**ASSIGNEE-INFORMATION:**

**NAME**

**COUNTRY**

**HITACHI LTD**

**N/A**

**APPL-NO: JP62192570**

**APPL-DATE: August 3, 1987**

**INT-CL\_(IPC): H03M001/82; H03K007/10**

**US-CL-CURRENT: 375/238,375/238**

**ABSTRACT:**

**PURPOSE:** To generate a PWM signal of high resolution without using a high frequency clock neither a long bit length counter by subjecting the resolution of a PWM signal generating means of low resolution to pulse width modulation furthermore in the period of PWM command data generation.

**CONSTITUTION:** The clock pulse from a clock pulse generator 1 is counted by a counter 2, and the counted value of the counter 2 and PWM command data  $D<SB>1</SB>$  are compared with each other by a digital comparator 3. Output pulses of the counter 2 are counted by a counter 4, and the counted value of the counter 4 and PWM command data  $D<SB>2</SB>$  are compared with each other by a digital comparator 5. Outputs of these digital comparators 3 and 5 are processed by a NOR gate 7 and a D type flip flop 6, and a PWM signal where the resolution of the PWM signal outputted from the digital comparator 3 is enhanced is outputted.

**COPYRIGHT: (C)1989,JPO&Japio**



⑨ 日本国特許庁(JP)

⑪ 特許出願公開

⑫ 公開特許公報(A)

昭64-37124

⑤ Int. Cl.

識別記号

庁内整理番号

④ 公開 昭和64年(1989)2月7日

H 03 M 1/82  
H 03 K 7/10

6832-5J  
7328-5J

審査請求 未請求 発明の数 1 (全8頁)

⑬ 発明の名称 パルス幅変調信号発生装置

⑭ 特 願 昭62-192570

⑮ 出 願 昭62(1987)8月3日

⑯ 発 明 者 久 保 謙 二 茨城県日立市久慈町4026番地 株式会社日立製作所日立研  
究所内  
⑯ 発 明 者 渡 辺 正 彦 茨城県日立市久慈町4026番地 株式会社日立製作所日立研  
究所内  
⑯ 発 明 者 大 前 力 茨城県日立市久慈町4026番地 株式会社日立製作所日立研  
究所内  
⑯ 発 明 者 羽 根 井 博 幸 千葉県習志野市東習志野7丁目1番1号 株式会社日立製  
作所習志野工場内  
⑰ 出 願 人 株式会社日立製作所 東京都千代田区神田駿河台4丁目6番地  
⑱ 代 理 人 弁理士 武 願 次 郎 外 1 名

明 細 書

1. 発明の名称

パルス幅変調信号発生装置

2. 特許請求の範囲

1. 第1のパルス幅変調指令データに対応したパルス幅比を持つ第1のパルス幅変調信号を発生する第1のパルス幅変調信号発生手段と、第2のパルス幅変調指令データに対応したパルス幅比を持ち、前記第1のパルス幅変調信号より長い周期の第2のパルス幅変調信号を発生する第2のパルス幅変調信号発生手段とを備え、前記第2のパルス幅変調信号に応じて、前記第1のパルス幅変調信号のパルス幅比を変化させることを特徴とするパルス幅変調信号発生装置。
2. 前記第2のパルス幅変調信号の周期を、パルス幅変調指令データの指令周期と同期させることを特徴とする特許請求の範囲第1項記載のパルス幅変調信号発生装置。

3. 発明の詳細な説明

(産業上の利用分野)

本発明は、デジタルデータをパルス幅変調(Pulse Width Moduration、以下PWMという)信号に変換する装置に係り、特に、簡単な回路構成で高分解能のPWM信号を得ることを可能としたパルス幅変調信号発生装置に関する。

(従来の技術)

マイクロプロセッサ等を用いるデジタル制御装置において、その制御出力をアナログ値に変換する手段として、DA(Digital to Analog)変換器を用いる方法と、制御出力を該制御出力に対応したパルス幅比のPWM信号に変換し、そのPWM信号をレベル変換した後、低域通過フィルタにより高周波成分を除去する方法とが知られている。前記2つの方法のうち、PWM信号に変換する方法は、2値レベルの信号として情報を伝送できること、フォトカプラ等を用いて信号を絶縁できること等の特徴を有し、モータのデジタル制御装置における速度や駆動トルクの指令方式に用いる方法として適している。

このようなPWM回路としては、高分解能のク

ロックパルスをカウンタで計数し、その計数値と制御出力データとをデジタル比較器で比較することにより、パルス幅を決定する方法が用いられる。このとき、クロックパルスの周波数を $f_{cx}$ 、制御出力データの分解を $1/N_c$ 、PWM信号の周波数を $f_c$ とすると、

$$f_c = f_{cx}/N_c \quad \text{----- (I)}$$

の関係が成立する。いま、 $f_{cx} = 20\text{MHz}$ 、 $f_c = 20\text{kHz}$ とすると、 $N_c = 1000$ となる。すなわち、この例では、制御出力データの分解能は、 $1/1000$ であり、符号付きとして考えると $\pm 1/500$ にしかならないことになる。PWM信号の周波数 $f_c$ は、制御出力の応答周波数より充分高く選定する必要があり、このため、PWM信号による制御出力データの分解能を向上させるには、クロックパルスの周波数 $f_{cx}$ を高くする方法が用いられる。しかし、この方法は、PWMのために用いるカウンタ等の動作速度の制限により、使用可能なクロックパルスの周波数 $f_{cx}$ にも上限がある。

と、該第1のPWM信号発生手段のPWM信号の発生周期より長い周期でPWMを行う第2のPWM信号発生手段とを備え、この第2のPWM信号に応じて第1のPWM信号のパルス幅比を、第2のPWM信号発生手段からの第2のPWM信号の分解能だけ変化させるようにすることにより達成される。

(作用)

第1のPWM信号発生手段は、PWM指令データの分解能より低い分解能で量子化された第1のPWM信号を発生する。これに対し、第2のPWM信号発生手段は、量子化データの剰余を第1のPWM信号発生手段の第1のPWM信号の周期の数倍の周期でPWM変調する。この第2のPWM信号発生手段による第2のPWM信号は、1量子化単位を更に高分解能化するための制御信号として利用される。第1のPWM信号発生手段からの第1のPWM信号は、前記第2のPWM信号発生手段の第2のPWM信号に基づいて、そのパルス幅が最小単位だけ変化せしめられる。これにより、

クロックパルスの周波数 $f_{cx}$ を高くすることなく、PWM信号による制御出力データの高分解能化を図るための従来技術として、例えば、特開昭60-190029号公報等に記載された技術が知られている。この種従来技術は、分解能の異なるPWM回路を組合わせて使用することにより、等価的に高分解能のPWM信号による制御出力データを得るものである。

(発明が解決しようとする問題点)

前記従来技術は、クロックパルスの周波数 $f_{cx}$ を高くすることなく、PWM信号の分解能を高くすることができるという利点を有するが、2つのPWM回路を組合わせて用いるため、回路構成が複雑となるという問題点があつた。

本発明の目的は、簡単に安価な回路構成で高分解能のPWM信号を発生することが可能なパルス幅変調信号発生装置を提供することにある。

(問題点を解決するための手段)

本発明によれば、前記目的は、所望の周波数のPWM信号を発生する第1のPWM信号発生手段

第2のPWM信号発生手段からの第2のPWM信号の1周期の平均として見ると、PWM信号の等価的なパルス幅比は、前述の2つのPWM信号発生手段の発生するPWM信号の分解能の積として制御されることになり、高い分解能のPWM信号が出力される。この場合、PWM信号の周期は、第1のPWM信号発生手段による周期であり、この第1のPWM信号発生手段による第1のPWM信号は、低分解能でよいため、クロックパルスを高周波化することなく、高い分解能のPWM信号を発生することができる。

(実施例)

以下、本発明によるパルス幅変調信号発生装置の一実施例を図面により詳細に説明する。

第1図は本発明の一実施例の構成図、第2図はその動作を説明する波形図である。第1図において、1はクロックパルス発生器、2、4はカウンタ、3、5はデジタル比較器、6はD型フリップフロップ、7はNORゲートである。

本発明によるパルス幅変調信号発生装置の一実

施例は、第1図に示すように、クロックパルス発生器1と、該クロックパルス発生器1からのクロックパルスを一定周期で計数するカウンタ2と、該カウンタ2の計数値と設定される第1のPWM指令データD<sub>1</sub>とを比較するデジタル比較器3とより成る第1のPWM信号発生手段、カウンタ2の出力パルスを計数し所定の周期毎にその計数値がリセットされるカウンタ4と、該カウンタ4の計数値と設定される第2のPWM指令データD<sub>2</sub>とを比較するデジタル比較器5とより成る第2のPWM信号発生手段、第1のPWM信号発生手段から出力されるPWM信号に対し、クロックパルス1周期分の遅延を挿入するD型フリップフロップ6及び遅延挿入の有無を、第2のPWM信号発生手段から出力される第2のPWM信号によつて制御するNORゲート7により構成される。

次に、第1図に示すパルス幅変調信号発生装置の動作を第2図(A)～第2図(F)に示す動作波形図を参照して説明する。

カウンタ2は、クロックパルス発生器1からの

第2図(A)に示すようなクロックパルスCLKを計数する。その計数値C<sub>2</sub>は、第2図(B)に示すように変化する。カウンタ2の計数周期T<sub>2</sub>は、図示パルス幅変調信号発生装置からのPWM信号によつて制御される被制御装置に必要とされるPWM信号の発生周期に設定される。いま、カウンタ2が、クロックパルスCLKを0からN<sub>1</sub>-1まで計数したとき、カウンタ2の計数周期T<sub>2</sub>になるとすれば、これらとクロックパルス周期T<sub>ck</sub>との関係は、次式で表わされる。

$$T_2 = N_1 \cdot T_{ck} \quad \text{----- (I)}$$

デジタル比較器3は、このカウンタ2の計数値C<sub>2</sub>と第1のPWM指令データD<sub>1</sub>とを比較し、第2図(C)に示すような第1のPWM信号S<sub>1</sub>を発生する。この第1のPWM信号S<sub>1</sub>の周波数f<sub>1</sub>は、カウンタ2の計数周波数1/T<sub>2</sub>と等しく、その分解能は、1/N<sub>1</sub>である。この第1のPWM信号S<sub>1</sub>の分解能は、必要とされる分解能より低い分解能でよく、第1のPWM指令データD<sub>1</sub>は、必要とする分解能を有する指令データの

例えば上位複数ビットが用いられる。

一方、カウンタ4は、カウンタ2の1周期毎のパルスを計数し、その計数値C<sub>4</sub>は、第2図(D)に示すように変化する。カウンタ4の計数周期T<sub>4</sub>は、PWM指令データの変更タイミング周期に設定される。カウンタ4が、その計数値を0からN<sub>2</sub>-1まで変化したとき、カウンタ4の計数周期T<sub>4</sub>になるとすれば、計数周期T<sub>4</sub>は、

$$T_4 = N_2 \cdot T_2 = N_2 \cdot N_1 \cdot T_{ck} \quad \text{----- (II)}$$

と表わされる。デジタル比較器5は、このカウンタ4の計数値C<sub>4</sub>と第2のPWM指令データD<sub>2</sub>とを比較し、第2図(E)に示すような第2のPWM信号S<sub>2</sub>を発生する。この第2のPWM信号S<sub>2</sub>は、周期T<sub>4</sub>の期間の平均として1/N<sub>2</sub>の分解能を有し、第1のPWM信号S<sub>1</sub>の周期T<sub>2</sub>の期間における等価的な分解能を向上させるための制御信号として用いられる。この場合の第2のPWM指令データD<sub>2</sub>は、必要とする分解能を有する指令データの例えば下位の複数ビットが用いられる。すなわち、第1図に示すPWM信号発生装置

に与えられるトータルの高分解能のPWM指令データは、必要とする分解能より低い、予め設定された分解能のPWM信号を与えるための第1のPWM指令データD<sub>1</sub>と、この第1のPWM指令データD<sub>1</sub>によつて得られる第1のPWM信号を高分解能化するための剰余データとしての第2のPWM指令データD<sub>2</sub>とに分離されて、デジタル比較器3、5に与えられる。

D型フリップフロップ6は、周波数f<sub>1</sub>の第1のPWM信号S<sub>1</sub>の立下り時点を1クロックパルス周期T<sub>ck</sub>だけ遅延させるために用いられる。D型フリップフロップ6のD端子に入力される第1のPWM信号S<sub>1</sub>の信号レベルは、CP端子に入力されるクロックパルスCLKの立上り時点でD型フリップフロップ6にラッチされ、1クロックパルス周期後、Q端子の信号レベルとして出力されるが、第1のPWM信号S<sub>1</sub>が"1"レベルとなつたとき、直ちに、Q端子の出力を"1"に立上げるため、第1のPWM信号S<sub>1</sub>は、D型フリップフロップのP端子にも接続されている。

第1のPWM信号 $S_1$ の立下り時点を1クロックパルス周期だけ遅延させるため、NORゲート7とD型フリップフロップ6のC端子を用い、1クロックパルス遅延の制御を第2のPWM信号 $S_2$ で行うことにより、第1のPWM信号 $S_1$ の分解能は、第2のPWM信号 $S_2$ の周期でさらに高分解能化される。いま、第2のPWM信号 $S_2$ の出力が“1”のとき、NORゲート7の出力は“0”となり、D型フリップフロップ6のC端子のクリア機能が働かず、第1のPWM信号 $S_1$ は、その立下りが1クロック遅延されてD型フリップフロップ6のQ端子より出力される。一方、第2のPWM信号 $S_2$ の出力が“0”のとき、第1のPWM信号 $S_1$ が“0”となれば、NORゲート7の出力は、直ちに“1”となつて、D型フリップフロップ6のC端子のクリア機能が働くため、第1のPWM信号 $S_1$ は、その立下りが1クロック遅延されることなく、そのままD型フリップフロップ6のQ端子より出力される。このように、第2のPWM信号 $S_2$ の信号により、第1のPWM信

号 $S_1$ のバース幅を1クロックパルス周期だけ変動することにより、周期 $T_1$ の平均として、第1のPWM信号 $S_1$ の分解能を更に $1/N_2$ だけ高分解能化することが可能である。この高分解能化されたPWM信号 $S_1$ は、D型フリップフロップ6のQ端子から第2図(F)に示すように得られる。

いま、第1のPWM信号 $S_1$ の分解能を $1/N_1$ 、第2のPWM信号 $S_2$ の分解能を $1/N_2$ とすると、最終的に得られるPWM信号 $S_1$ の分解能 $1/N_c$ は、次式で表わされる。

$$1/N_c = 1/(N_1 \cdot N_2) \quad \text{-----(3)}$$

また、このPWM信号 $S_1$ の周波数 $f_c$ は、第1のPWM信号の周波数 $f_1$ に等しい。例えば、 $N_1 = 256$ 、 $f_1 = 20\text{kHz}$ 、第2のPWM信号の周期 $T_2 = 1\mu\text{s}$ とすると、クロックパルス周波数 $f_{ck}$ は、

$$f_{ck} = N_1 \cdot f_1 = 5\text{MHz}$$

となり、また、第2のPWM信号の分解能 $1/N_2$ は、

$1/N_2 = T_1/T_2 = 1/20$   
となる。周期 $T_2 (= 1\mu\text{s})$ の期間のPWM信号 $S_2$ の平均的な分解能 $1/N_c$ は、

$1/N_c = 1/(N_1 \cdot N_2) \approx 1/5000$   
となり、クロックパルス周波数が $5\text{MHz}$ とそれほど高い周波数ではないにもかかわらず、PWM信号 $S_1$ を高分解能化できる。しかも、このPWM信号 $S_1$ の周波数 $f_c$ は、 $20\text{kHz}$ と十分に高く、例えば、このPWM信号 $S_1$ をRC回路のような簡単な平滑フィルタを用いてアナログ信号に変換する場合、フィルタのカットオフ周波数を $1\text{kHz}$ 程度に設定すれば、PWM信号のリップル分をほぼ完全に除去したアナログ信号を得ることができる。このとき、 $1\text{kHz}$ 以下の信号成分は、減衰することなくアナログ値として得られるので、周期 $T_2 (= 1\mu\text{s})$ 間の平均として、高分解能であればよい信号情報は、十分に伝送可能である。

前述したように、第1図、第2図により説明した本発明の実施例によれば、高い周波数のクロッ

クパルスを用いなくても、比較的高い周波数のPWM信号を高分解能に制御できるので、モータ等のデジタル制御における速度指令回路、電流指令回路等を安価な回路で実現でき、その回路構成も低分解能のカウントを組合わせて実現でき、回路構成を簡略化できる。

第3図は本発明の第2の実施例の構成図、第4図はその動作を説明する波形図である。第3図において、8はマイクロプロセッサ、9はダウンカウンタ、10はフリップフロップであり、他の符号は第1図の場合と同一である。

第3図に示す第2の実施例は、第1図に示すカウンタ4に代り、ダウンカウンタ9が備えられた点、及びマイクロプロセッサ8、フリップフロップ10が追加された点で、第1図に示す実施例の構成と相違するが、その他の構成は同一である。

マイクロプロセッサ8は、第4図(A)に示す周期 $T_c$ の制御演算同期を持ち、この制御演算同期のタイミング毎に、第1図の実施例の場合と同一の第1及び第2のPWM指令信号 $D_1$ 、 $D_2$ を発

生ずる。第1の指令信号により第1のPWM信号S<sub>1</sub>を発生するクロックパルス発生器1、カウンタ2、デジタル比較器3より成る第1のPWM信号発生手段は、第1図の場合と全く同様に動作する。このときのカウント2の動作と第1のPWM信号S<sub>1</sub>が第4図(B)、(C)に示されており、これは、第2図(B)、(C)の場合と同一である。

ダウンカウンタ9には、マイクロプロセッサ8から制御演算周期毎に第2のPWM指令データD<sub>2</sub>がプリセットされ、ダウンカウンタ9は、その後カウンタ2の計数周期毎に、プリセットされた指令データのダウンカウントを行い、アンダーフローが生じたときにボロー信号を出力する。このダウンカウンタ9の計数の様子が第4図(D)に示されている。フリップフロップ10は、マイクロプロセッサ8の制御演算周期T<sub>c</sub>毎にセットされ、ダウンカウンタ9からのボロー信号によりリセットされる動作を周期T<sub>c</sub>毎に繰返すことにより、第2のPWM指令データD<sub>2</sub>に基づいた第2のP

WM信号S<sub>2</sub>を第4図(E)に示すように出力する。すなわち、第3図に示す本発明の第2の実施例は、ダウンカウンタ9とフリップフロップ10とにより第2のPWM信号発生手段が構成されている。

D型フリップフロップ6とNORゲート7とによる回路が、第2のPWM信号S<sub>2</sub>に基づいて、第1のPWM信号S<sub>1</sub>に対し、1クロックパルスの遅延を与える制御を実行し、第1のPWM信号S<sub>1</sub>を高分解能化したPWM信号S<sub>2</sub>を出力することは、第1図により説明した場合と全く同様であり、その出力のPWM信号S<sub>2</sub>が第4図(F)に示されている。

この第2の実施例は、第4図の動作波形から明らかなように、第2のPWM信号S<sub>2</sub>の周期T<sub>2</sub>と、マイクロプロセッサ8のPWM指令周期、すなわち制御演算周期T<sub>c</sub>とが一致しており、この1演算周期内の第1のPWM信号S<sub>1</sub>を、第2のPWM信号S<sub>2</sub>でパルス幅変調することにより、この1演算周期内で平均的に高分解能化したP

WM信号S<sub>2</sub>を得ることができる。このときのPWM信号S<sub>2</sub>の分解能1/N<sub>2</sub>は、第1の実施例の場合と同様に、

$$1/N_2 = 1/(N_1 \cdot N_2)$$

である。ここで、1/N<sub>1</sub>は第1のPWM信号の分解能、1/N<sub>2</sub>は制御演算周期T<sub>c</sub>内の第1のPWM信号数の逆数、すなわち、第2のPWM信号S<sub>2</sub>の分解能である。また、第1のPWM信号S<sub>1</sub>の周期をT<sub>1</sub>とすれば、

$$1/N_2 = T_1/T_c$$

となる。従つて、例えば、T<sub>c</sub> = 1 μs、f<sub>1</sub> = 1/T<sub>1</sub> = 20 KHzとすれば、N<sub>2</sub> = 20となる。第3図に示す本発明の実施例は、第1図に示す実施例の場合と同様に、第1のPWM信号S<sub>1</sub>の周期T<sub>1</sub>(=T<sub>c</sub>)の期間における平均的な分解能を更に20倍とすることができる。

前述した本発明の第2の実施例によれば、第2のPWM信号S<sub>2</sub>の周期をマイクロプロセッサ8のPWM指令データ発生周期T<sub>c</sub>に同期させることができるので、第2のPWM信号S<sub>2</sub>の周期T<sub>2</sub>

をPWM指令データ発生周期T<sub>c</sub>まで長くすることができる。これにより、N<sub>2</sub> = T<sub>c</sub>/T<sub>1</sub>の値を大きくすることができるので、PWM信号S<sub>2</sub>をより高分解能の信号にすることができる。

第5図は本発明の第3の実施例を示す構成図、第6図はその動作を説明する波形図である。第5図における符号は、第3図の場合と同一である。

第5図に示す本発明の第3の実施例は、ダウンカウンタ9のボロー信号をマイクロプロセッサ8に対する割込信号として用いるものであり、第3図に示すD型フリップフロップ6、NORゲート7及びフリップフロップ10の機能をマイクロプロセッサ8に行わせるように構成されている。第6図(A)、(B)、(C)、(E)に示すこの実施例の動作波形は、第4図(A)、(B)、(C)、(F)に示す動作波形と同一であり、第6図(D)に示す割込信号INTは、ダウンカウンタ9がアンダーフローしたときのボロー信号である。

この実施例におけるマイクロプロセッサ8は、ダウンカウンタ9のボロー信号による割込信号I

NTにより、第1のPWM信号に対する第1の指令データD<sub>1</sub>を変化させて設定することにより、前述した第1、第2の実施例において、ハードウェアにより行っていた、第1のPWM信号の立下り時の1クロックパルスの遅延制御を、ソフトウェア処理で実行するものである。

すなわち、マイクロプロセッサ8は、マイクロプロセッサ8の指令周期T<sub>c</sub>毎に第1及び第2のPWM指令データD<sub>1</sub>、D<sub>2</sub>を演算出力する。このとき、第1のPWM信号発生手段に対する第1のPWM指令データの設定値(第5図の場合、デジタル比較器3への設定値)は、第1のPWM指令データD<sub>1</sub>に+1加算をした値を用い、第2のPWM信号発生用のダウンカウンタ9には、第2のPWM指令データD<sub>2</sub>をそのまま設定する。ダウンカウンタ9が設定値D<sub>2</sub>をダウンカウントし、ボロー信号を出力すると、そのボロー信号は、マイクロプロセッサ8に対する割込信号INTとしてマイクロプロセッサ8に与えられる。マイクロプロセッサ8は、この割込信号INTが与えられ

ると、第1のPWM指令信号を最初の設定値から1だけ減算した値、すなわちD<sub>1</sub>として再設定する。これにより、デジタル比較器3は、マイクロプロセッサ8の制御演算周期すなわち指令周期T<sub>c</sub>内で高分解能化されたPWM信号S<sub>1</sub>を直接出力する。

前述した本発明の第3の実施例によれば、遅延回路をハードウェアで構成する必要がなく、より簡単な回路構成で、高分解能のPWM信号を発生することができる。

前述の第1～第3の本発明の実施例は、PWM信号発生回路をデジタル比較器により構成しているが、本発明におけるPWM信号発生回路は、このような回路構成に限定される必要はなく、例えば、カウンタを用いた構成のもの等、どのような回路構成のものであつてもよい。

#### (発明の効果)

以上説明したように、本発明によれば、比較的分解能の低いPWM信号発生手段の分解能を、PWM指令データ発生周期内で更にパルス幅変調する

ことにより、高い周波数のクロックパルスや長いビット長のカウンタを用いることなく、高い周波数で高分解能のPWM信号を発生することが可能となる。このため、本発明によれば、モータ等のデジタルサーボ系におけるデータ指令回路のように、フォトカプラ等を用いた絶縁を行うためにPWM信号による指令が不可欠な応用技術では、特に適した装置を提供することができる。また、本発明は、回路構成が簡単であるため、ロボットのサーボ系等のような多軸のサーボアンプに対する速度や電流指令回路としても適している。

#### 4. 図面の簡単な説明

第1図は本発明の一実施例の構成図、第2図はその動作を説明する波形図、第3図は本発明の第2の実施例の構成図、第4図はその動作を説明する波形図、第5図は本発明の第3の実施例の構成図、第6図はその動作を説明する波形図である。

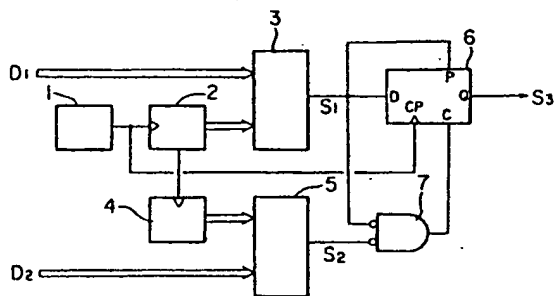
1……クロックパルス発生器、2、4……カウンタ、3、5……デジタル比較器、6……D型フリップフロップ、7……NORゲート、8……

マイクロプロセッサ、9……ダウンカウンタ、10……フリップフロップ。

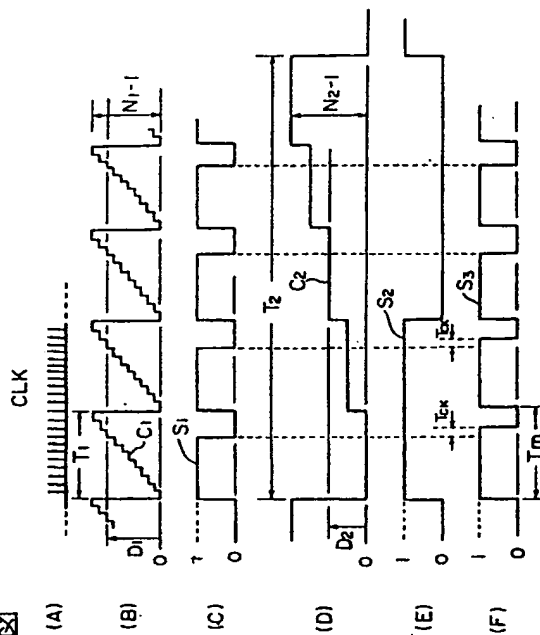
代理人 弁理士 武 頭次郎 (外1名)



第1図

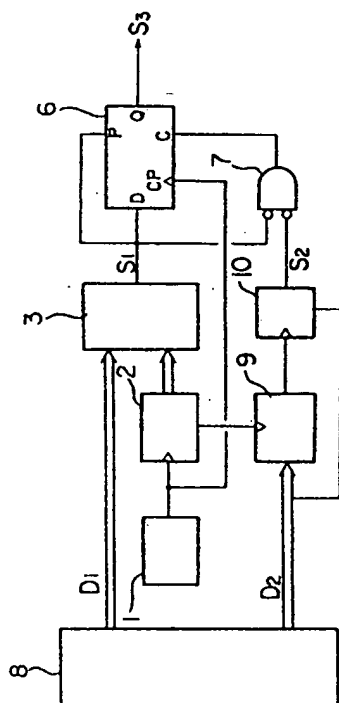


- 1 クロックパルス発生器
- 2 カウンタ
- 3 デジタル比較器
- 4 カウンタ
- 5 デジタル比較器
- 6 D型フリップフロップ
- 7 NOR ゲート



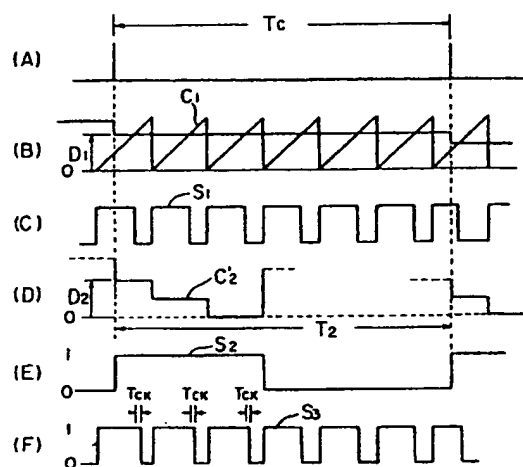
第2図

第3図

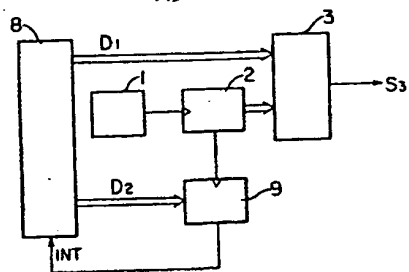


- 8 マイクロプロセッサ
- 9 ダウンカウンタ

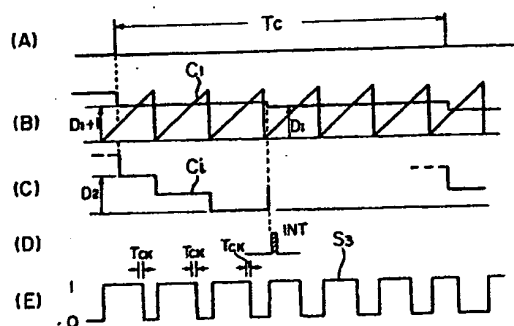
第4図



第5図



第6図





**This Page is Inserted by IFW Indexing and Scanning  
Operations and is not part of the Official Record**

**BEST AVAILABLE IMAGES**

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

- ☐ BLACK BORDERS
- ☐ IMAGE CUT OFF AT TOP, BOTTOM OR SIDES
- ☒ FADED TEXT OR DRAWING
- ☐ BLURRED OR ILLEGIBLE TEXT OR DRAWING
- ☐ SKEWED/SLANTED IMAGES
- ☐ COLOR OR BLACK AND WHITE PHOTOGRAPHS
- ☐ GRAY SCALE DOCUMENTS
- ☒ LINES OR MARKS ON ORIGINAL DOCUMENT
- ☐ REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY
- ☐ OTHER: \_\_\_\_\_

**IMAGES ARE BEST AVAILABLE COPY.**

**As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.**

**THIS PAGE BLANK (USPTO)**